```
entry:
                                                                 call void @llvm.dbg.value(metadata ptr %a, metadata !21, metadata
                                                                 ... !DIExpression()), !dbg !27
                                                                 call void @llvm.dbg.value(metadata ptr %b, metadata !22, metadata
                                                                 ...!DIExpression()), !dbg!27
                                                                 call void @llvm.dbg.value(metadata ptr %c, metadata !23, metadata
                                                                 ...!DIExpression()), !dbg!27
                                                                 call void @llvm.dbg.value(metadata i32 %n, metadata !24, metadata
                                                                 ...!DIExpression()), !dbg !27
                                                                 tail call void asm sideeffect ".inst 0x2520e020", ""() #7, !dbg !28, !srcloc
                                                                 call void @llvm.dbg.value(metadata i32 0, metadata !25, metadata
                                                                 .. !DIExpression()), !dbg !31
                                                                 %cmp11 = icmp sgt i32 %n, 0, !dbg !32
                                                                 br i1 %cmp11, label %for.body.preheader, label %for.cond.cleanup, !dbg !34
                                                         for.body.preheader:
                                                         %wide.trip.count = zext i32 %n to i64, !dbg !32
                                                         %0 = call i32 @llvm.vscale.i32(), !dbg !34
                                                          %extended.vscale = zext i32 %0 to i64
                                                         %1 = shl i64 %extended.vscale, 2, !dbg !34
                                                         %2 = icmp uge i64 %wide.trip.count, %1, !dbg !34
                                                         br i1 %2, label %Pre.Vectorization, label
                                                         ... %Preheader.for.remaining.iterations, !dbg !34
          Pre. Vectorization:
          \%5 = \text{call} < \text{vscale x 4 x i64} > \text{@llvm.experimental.stepvector.nxv4i64}()
          \%6 = \text{call i32 @llvm.vscale.i32()}
          %extended.vscale1 = zext i32 %6 to i64
          %step.value = shl i64 %extended.vscale1, 2
          %7 = urem i64 %wide.trip.count, %step.value
          %total.iterations.to.be.vectorized = sub i64 %wide.trip.count, %7
          %8 = insertelement <vscale x 4 x i64> poison, i64 %step.value, i64 0
          %stepVector.update.values = shufflevector <vscale x 4 x i64> %8, <vscale x 4
          ... x i64> poison, <vscale x 4 x i32> zeroinitializer
          br label %vectorizing.block
         vectorizing.block:
         %9 = phi i64 [ 0, %Pre.Vectorization ], [ %20, %vectorizing.block ]
          \%10 = \text{phi} < \text{vscale x } 4 \text{ x } i64 > [\%5, \%Pre. Vectorization], [\%21,
          ... %vectorizing.block ]
          %11 = and <vscale x 4 x i64> %10, shufflevector (<vscale x 4 x i64>
         ... insertelement (<vscale x 4 x i64> poison, i64 1, i64 0), <vscale x 4 x i64>
         ... poison, <vscale x 4 x i32> zeroinitializer)
          %12 = icmp eq < vscale x 4 x i64 > %11, zeroinitializer
          %13 = bitcast < vscale x 4 x i1 > %12 to < vscale x 4 x i1 >
          %negated.vector = xor <vscale x 4 x i1> %13, shufflevector (<vscale x 4 x
          ... i1> insertelement (<vscale x 4 x i1> poison, i1 true, i32 0), <vscale x 4 x
         ... i1> poison, <vscale x 4 x i32> zeroinitializer)
          %14 = getelementptr inbounds i32, ptr %a, i64 %9, !dbg !43
          %15 = getelementptr inbounds i32, ptr %b, i64 %9, !dbg !49
          %16 = getelementptr inbounds i32, ptr %c, i64 %9, !dbg !51
          %17 = call <vscale x 4 x i32> @llvm.aarch64.sve.ld1.nxv4i32(<vscale x 4 x
          ... i1> %negated.vector, ptr %14)
         \%18 = \text{call} < \text{vscale x 4 x i} 32 > \text{@llvm.aarch} 64.\text{sve.ld1.nxv4} i 32 (< \text{vscale x 4 x})
          ... i1> %negated.vector, ptr %15)
          %19 = call <vscale x 4 x i32> @llvm.aarch64.sve.mul.nxv4i32(<vscale x 4 x
         ... i1> %negated.vector, \langle vscale \times 4 \times i32 \rangle \%18, \langle vscale \times 4 \times i32 \rangle \%17)
          call void @llvm.aarch64.sve.st1.nxv4i32(<vscale x 4 x i32> %19, <vscale x 4
          \dots x i1> %negated.vector, ptr %16)
         %20 = add i64 %step.value, %9
         %21 = add <vscale x 4 x i64> %10, %stepVector.update.values
         %terminate.condition = icmp uge i64 %20, %total.iterations.to.be.vectorized
          br i1 %terminate.condition, label %middle.block, label %vectorizing.block
middle.block:
%condition = icmp eq i64 %7, 0
br i1 %condition, label %for.cond.cleanup.loopexit, label
... %Preheader.for.remaining.iterations
                                           F
                    Preheader.for.remaining.iterations:
                     %22 = phi i64 [ 0, %for.body.preheader ], [ %20, %middle.block ]
                     br label %for.body
              for.body:
              %indvars.iv = phi i64 [ %indvars.iv.next, %for.inc ], [ %22,
              ... %Preheader.for.remaining.iterations ]
              call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !25, metadata
              ... !DIExpression()), !dbg !31
%rem15 = and i64 %indvars.iv, 1, !dbg !39
              %cmp1.not = icmp eq i64 %rem15, 0, !dbg !39
              br i1 %cmp1.not, label %for.inc, label %if.then, !dbg !42
                                       if.then:
                                        %arrayidx = getelementptr inbounds i32, ptr %a, i64 %indvars.iv, !dbg !43
                                        %3 = load i32, ptr %arrayidx, align 4, !dbg !43, !tbaa !45
                                        %arrayidx3 = getelementptr inbounds i32, ptr %b, i64 %indvars.iv, !dbg !49
                                        %4 = load i32, ptr %arrayidx3, align 4, !dbg !49, !tbaa !45
                                        %mul = mul nsw i32 %4, %3, !dbg !50
                                        %arrayidx5 = getelementptr inbounds i32, ptr %c, i64 %indvars.iv, !dbg !51
                                        store i32 %mul, ptr %arrayidx5, align 4, !dbg !52, !tbaa !45
                                        br label %for.inc, !dbg !53
              %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !54
              call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !25,
              ... metadata !DIExpression()), !dbg !31
              %exitcond.not = icmp eq i64 %indvars.iv.next, %wide.trip.count, !dbg !32
              br i1 %exitcond.not, label %for.cond.cleanup.loopexit, label %for.body, !dbg
              ...!34,!llvm.loop!55
              for.cond.cleanup.loopexit:
              br label %for.cond.cleanup, !dbg !35
                                      for.cond.cleanup:
                                      tail call void asm sideeffect ".inst 0x2520e040", ""() #7, !dbg !35, !srcloc
                                       ret void, !dbg !38
```